Continuous-time segmentation networks

John G. Harris

John G. Harris, "Continuous-time segmentation networks," Proc. SPIE 1473, Visual Information Processing: From Neurons to Chips, (9 July 1991); doi: 10.1117/12.45549

SPIE.

Event: Orlando '91, 1991, Orlando, FL, United States

CONTINUOUS-TIME SEGMENTATION NETWORKS

John G. Harris

Computation and Neural Systems Program, 216-76
California Institute of Technology
Pasadena, CA, 91125

and

Hughes Artificial Intelligence Center
3011 Malibu Canyon Rd.
Malibu, CA 90265

ABSTRACT

Segmentation is a basic problem in computer vision. We introduce the tiny-tanh network, a continuous-time network that segments scenes based upon intensity, motion, or depth. The tiny-tanh algorithm maps naturally to analog circuitry since it was inspired by previous experiments with analog VLSI segmentation hardware. A convex Lyapunov energy is utilized so that the system does not get stuck in local minima. No annealing algorithms of any kind are necessary—a sharp contrast to previous software/hardware solutions of this problem.

1 INTRODUCTION

A large class of vision algorithms is based on minimizing an associated cost functional. Such a variational formalism is attractive because it allows a priori constraints to be stated explicitly. The single most important constraint is that the physical processes underlying image formation—such as depth, orientation, and surface reflectance—change slowly in space. For instance, the depths of neighboring points on a surface are usually similar. Standard regularization algorithms embody this smoothness constraint and lead to quadratic variational functionals with a unique, global minimum (Horn and Schunck, 1981; Hildreth, 1984; Poggio, Torre and Koch, 1985; Poggio, Voorhees and Yuille, 1986; Grimson, 1981). These quadratic functionals can be mapped onto linear resistive networks, such that the stationary voltage distribution, corresponding to the state of least power dissipation, is equivalent to the solution of the variational functional (Horn, 1974; Poggio and Koch, 1985). Inputs are supplied by injecting currents into the appropriate nodes.

0-8194-0582-5/91/$4.00

Smoothness breaks down, however, at discontinuities caused by occlusions or by differences in the physical processes underlying image formation (such as different surface reflectance properties). Detecting these discontinuities becomes crucial, not only because otherwise smoothness is applied incorrectly but also because the locations of discontinuities are often required for further image analysis and understanding (for example, we can often find the outline of a moving object reliably by detecting discontinuities in the optical flow field). Geman and Geman (1984) first introduced a class of stochastic algorithms, based on Markov random fields, that explicitly encodes the absence or presence of discontinuities by means of binary variables. Their approach was extended and modified by numerous researchers to account for discontinuities in depth, texture, optical flow, and color. An appropriate energy functional is minimized using stochastic optimization techniques, such as simulated annealing. Various deterministic approximations, based on continuation methods or on mean field theory yield next-to-optimal solutions (Koch, Marroquin and Yuille, 1986; Terzopoulos, 1986; Blake and Zisserman, 1987; Hutchinson *et al.*, 1988; Blake, 1989; Geiger and Girosi, 1989).

In the 1-D case the sparse and noisy depth data d_i are given on a discrete grid. Associated with each lattice point is the value of the recovered surface u_i and a binary line discontinuity ℓ_i. When the surface is expected to be smooth except at isolated discontinuities, the functional to be minimized is given by

$$J(u,\ell) = \sum_i \left[(d_i - u_i)^2 + \lambda^2(u_{i+1} - u_i)^2(1 - \ell_i) + \alpha\ell_i\right] \tag{1}$$

where λ and α are free parameters. The first term, with the sum including only those locations i where data exist, forces the surface u to be close to the measured data d. The second term implements the piecewise smooth constraint: if all variables, with the exception of u_i, u_{i+1}, and ℓ_i, are held fixed and $\lambda^2(u_{i+1} - u_i)^2 < \alpha$, then it is "cheaper" to pay the price $\lambda^2(u_{i+1} - u_i)^2$ and to set $\ell_i = 0$ than to pay the larger price α; if the gradient becomes too steep, $\ell_i = 1$, and the surface is segmented at that location. The final surface u is the one that best satisfies the conflicting demands of piecewise smoothness and fidelity to the measured data.

2 THE RESISTIVE FUSE

Analog circuits provide an interesting and powerful computational medium for developing early vision algorithms. As these small, inexpensive, low-power chips become common, researchers will possess real-time prototyping capabilities to address fundamental issues. Mead (1989) has pioneered the use of subthreshold analog CMOS for early sensory processing. For a review of analog circuits applied to early vision, see Horn (1989) and Koch (1989).

Instead of resorting to stochastic search techniques to find the global minimum of Eq. 1, we use a determin-

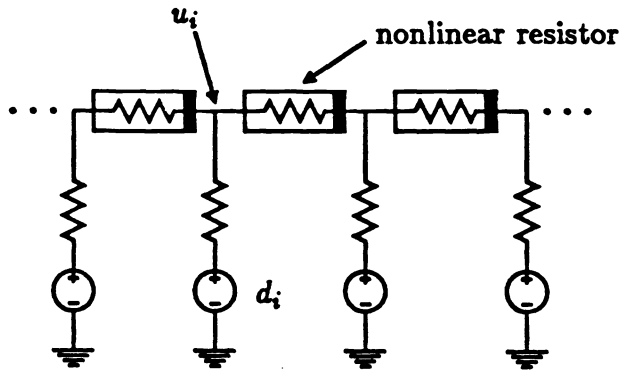

Figure 1: Schematic diagram for our hardware segmentation networks. A mesh of nonlinear resistive elements (shown as resistors inside rectangles) provides the smoothing and segmentation ability of the network. The data are given as battery values d_i with a unit conductance connecting each battery to the grid. In the absence of any discontinuities, the nonlinear resistors have a conductance of λ^2. The output is the voltage u_i at each node.

istic approximation and map the functional J onto the circuit shown in Fig. 1. The stationary voltage at every gridpoint then corresponds to u_i. If data exist at location i, a battery is set to d_i. The conductance between the battery and the grid is assumed to be 1 where data exist and 0 if no data exist at a particular location i. In the absence of any discontinuities (all $\ell = 0$), smoothness is implemented via a conductance of value λ^2 connecting neighboring grid points; that is, the nonlinear resistors in Fig. 1 can simply be considered linear resistors. The cost functional J can then be interpreted as the power dissipated by the circuit. If parasitic capacitances are added to the circuit, J acts as a Lyapunov function of the system and the stationary voltage distribution corresponds to the smooth surface.

Previously, we designed a two-terminal nonlinear device, which we call a resistive fuse, to implement piecewise smoothness (Harris *et al.*, 1990a). If the magnitude of the voltage drop across the device is less than $V_T = (\alpha^{1/2}/\lambda)$, the current through the device is proportional to the voltage, with a conductance of λ^2. This implements smoothness. If V_T is exceeded, the fuse breaks and the current goes to zero. Unlike the common electrical fuses in houses, the operation of the resistive fuse is fully reversible. We use an analog fuse with the I-V curve shown in Fig. 2, implementing a continuous version of the binary line discontinuities[1]. If the internal dynamics of the resistive fuse can be neglected, then it can be proven that our network will not oscillate, but rather will settle into a local minimum. The associated Lyapunov function is the electrical co-content (Harris *et al.*, 1989).

We built a 20 by 20 pixel VLSI chip, using the subcircuit types and design practices developed by Mead (1989). The slope λ^2 and the voltage threshold of all fuses can be set by off-chip voltage inputs. Figure 3a

[1]Perona and Malik (1989) discuss the performance of similar devices using anisotropic diffusion.

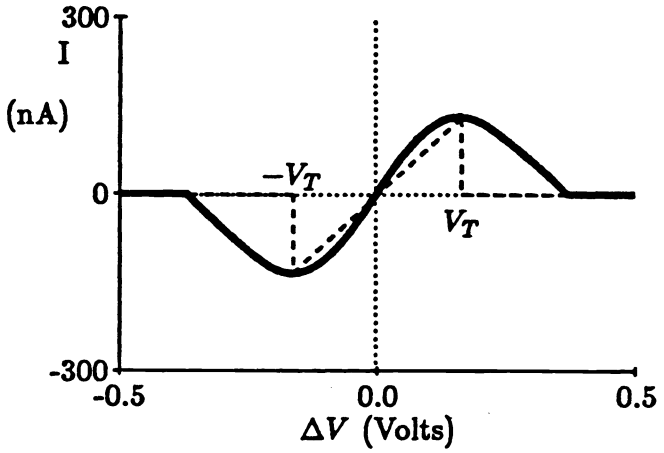

Figure 2: The solid line shows the measured current-voltage relationship for the resistive fuse. The I-V curve of a binary fuse is also illustrated. For a voltage of less than V_T across this two-terminal device, the circuit acts as a resistor with conductance λ^2. Above V_T, the current is either abruptly set to zero (binary fuse) or smoothly goes to zero (analog fuse). Independent voltage control lines allow us to change both λ^2 (over four orders of magnitude) and V_T (from 50 to 500 mV). In addition, we can vary the I-V curve continuously from the hyperbolic tangent of Mead's saturating resistor (HRES) to that of an analog fuse.

shows a figure eight pattern that was scanned into the chip. The height of the signal was 0.5V with evenly-distributed additive noise of $\pm$0.25V. Figure 3b shows the measured voltage values that were scanned off the chip. These voltages could represent image intensity, depth, or motion. Algorithm simulations have shown that the resistive fuse idea can be combined successfully with a motion algorithm to compute segmented optical flow fields from digitized images (Harris *et al.*, 1990b).

Mapping nonlinear computations to analog VLSI does not avoid the problem of local minima. Several continuation methods have been developed that gradually warp the Lyapunov energy of the system from a convex one to the final desired energy (Harris *et al.*, 1989, Lumsdaine *et al.*, 1990). Continuation methods rely on the fact that the unique minima of the convex energy are close to the global minima of the desired energy. Unfortunately, these methods require clocking circuitry to vary an on-chip voltage through a range of values. Furthermore, since our analog technology naturally implements continuous-time systems with on-chip photoreceptors, it would be awkward to resort to a sampled-time system. This problem is especially evident when we try to recover the optical flow in the presence of motion discontinuities (Harris *et al.*, 1990b). Motion already contains a natural time scale that we would have to ensure is much longer than the time scale of our continuation methods.

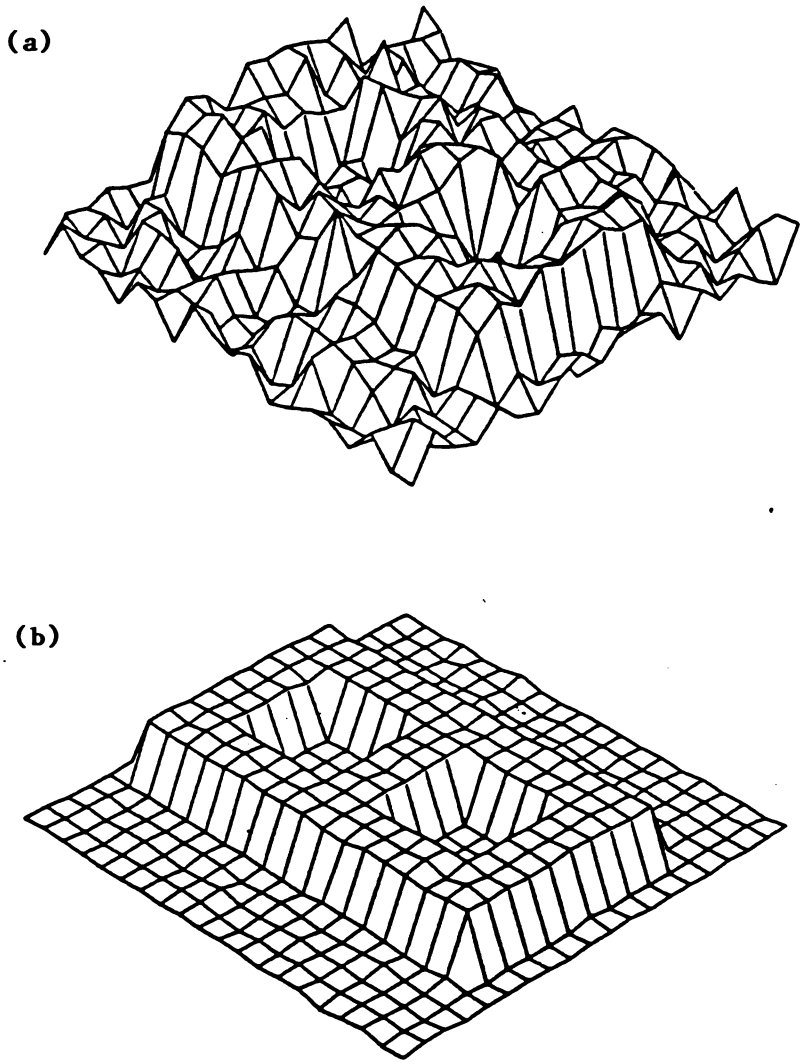

Figure 3: Experimental data from the two-dimensional resistive fuse chip. **(a)** A noisy figure eight pattern was used as input. The height of the signal is 0.5V with evenly distributed additive noise of ±0.25V. **(b)** The measured voltage output scanned off the chip.

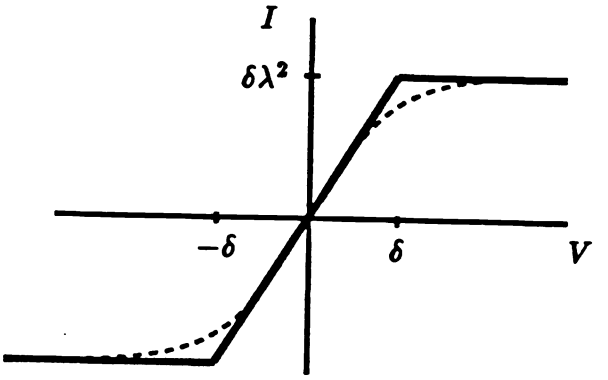

Figure 4: The desired saturating I-V characteristic is shown as a dotted line. Its piecewise linear approximation (solid line) was used for simulation and analysis. The slope of the linear region (λ^2) and the saturation threshold (δ) are adjustable parameters.

3 THE TINY TANH NETWORK

Rather than deal with resistive networks that have many possible stable states, we propose to utilize a network that has a single unique stable state. Any circuit made of independent voltage sources and two-terminal resistors with strictly increasing I-V characteristics has a single unique stable state. This result has been well known; a proof is given in Chua *et al.* (1987). Gradient descent algorithms are guaranteed to find this solution. Figure 4 shows a plot of our proposed element's I-V characteristic, a simple hyperbolic tangent relationship with an adjustable width and slope. The slope in the linear region is defined to be λ^2, and the curve saturates for voltage drops greater than δ. Since the derivative of the tanh function is always positive, the nonlinear resistive network containing them is guaranteed to converge to a unique minimum. A piecewise linear approximation of this curve was used in simulations and analysis. This saturating element is used as a the nonlinear resistive element shown in Figure 1.

Mead has constructed a saturating resistor in subthreshold analog VLSI (Mead, 1989). This device is often called HRES (Horizontal RESistor) since it was originally designed to model the horizontal cells in the mammalian retina. To evaluate how the saturating resistor performs, a fourth-order Runge-Kutta simulation was developed for a two-dimensional network of saturating resistors. A unit step edge was generated for a 20 x 20 pixel image and gaussian noise with $\sigma = .2$ was added. The filled circles in Figure 5 represent a 1-D cross section through this array. Figure 5a shows the typical segmentation result for the saturating resistor, $\delta = .1$ and $\lambda^2 = 4.0$. As Mead has observed, a network of saturating resistors performs an edge enhancement (Mead, 1989). Unfortunately, the noise is still evident in the output, and the curves on either side of the step have started to slope towards one another. As λ is increased to smooth out the noise, the two sides of the step will blend together into one homogeneous region. These same results are observed with hardware resistive networks using Mead's saturating resistor. These saturating resistors

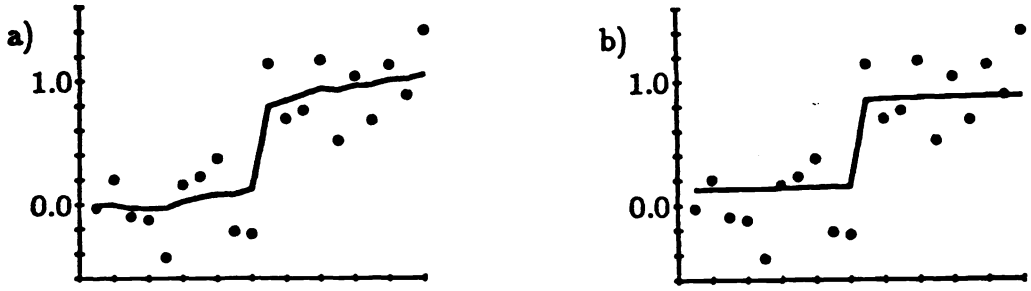

Figure 5: A 20x20 pixel unit step edge was generated, and gaussian noise with $\sigma = .2$ was added. The filled-circles indicate a 1-D cross section through this input array. **(a)** Shows the segmentation result for the saturating resistor in the usual setting, $\delta = .1$ and $\lambda = 2.0$. For these parameter settings, the noise is still visible in the output and the curves on either side of the step have started to slope towards one another. **(b)** The enhanced region of operation ($\delta = .005$ and $\lambda^2 = 256$) segments the input into two distinct uniform regions.

cannot segment data into regions of roughly uniform voltage.

The new idea proposed here is that the saturating resistor can be run in a very different region of operation where network segmentation properties are greatly enhanced. If we decrease the width of the linear region significantly ($\delta = .005$) while simultaneously increasing the conductance ($\lambda = 16$), the result shown in Fig. 5b is produced. The height of the step[2] detected is 200 times larger than δ. Figure 6 shows a simulation of the tiny-tanh network segmenting the famous mandrill image for various values of δ. The network segments the image into blocks of approximately uniform brightness with δ determining the scale of the computation. No continuation method or annealing strategy of any sort was necessary to produce these simulation results.

We can understand why the network performs so well by studying the step response of a 1D network. Figure 7a shows the input of a step to the network of step size h. The nodes labeled V_L and V_H are the nodes to the immediate left and right of the step. If we use the piecewise linear approximation to the tiny-tanh, there are two cases to consider for the ideal step input. The resistive element between nodes V_L and V_H is either in its linear or saturation region of operation. For small input steps, the network is effectively linear. For $\lambda \gg 1$, we find that $V_L \approx V_H \approx h/2$. For larger inputs, one resistive element can be replaced by a constant current source of value $\delta\lambda^2$. For $\lambda \gg 1$, we find that $V_L \approx \delta\lambda$ and $V_H \approx h - \delta\lambda$. By setting the voltages in the two regimes equal we find that the breakpoint occurs at $h \approx 2\delta\lambda$. Figure 7b depicts the step response of the tiny-tanh network.

[2]If the signal size was 100mV, for example, a resistor with a linear region with a total width of 1mV would be required to provide the same performance. Mead's saturating resistor has a minimum linear region width of about 100mV.

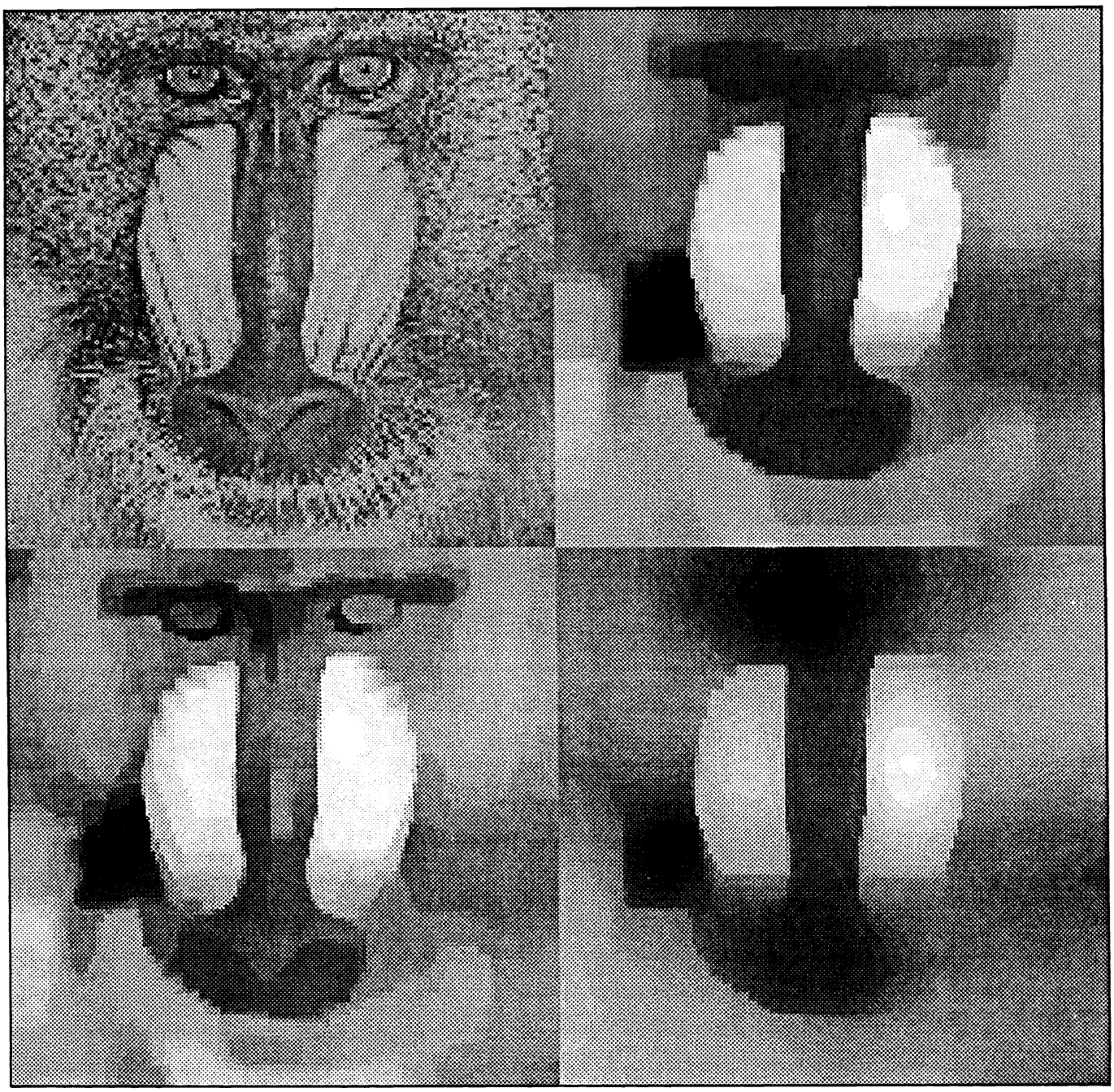

Figure 6: Simulations of tiny-tanh network on the mandrill image. $\lambda = 32$ for all four images. Top left is the original ($\delta = 0$), bottom left $\delta = 0.064$, top right $\delta = 0.128$ and bottom right $\delta = 0.512$. The original image consisted of 256x256 8-bit pixels. Each pixel ranged in value from 0 to 255. The network segmented the image into blocks of approximately uniform brightnes with δ determining the scale of the computation.

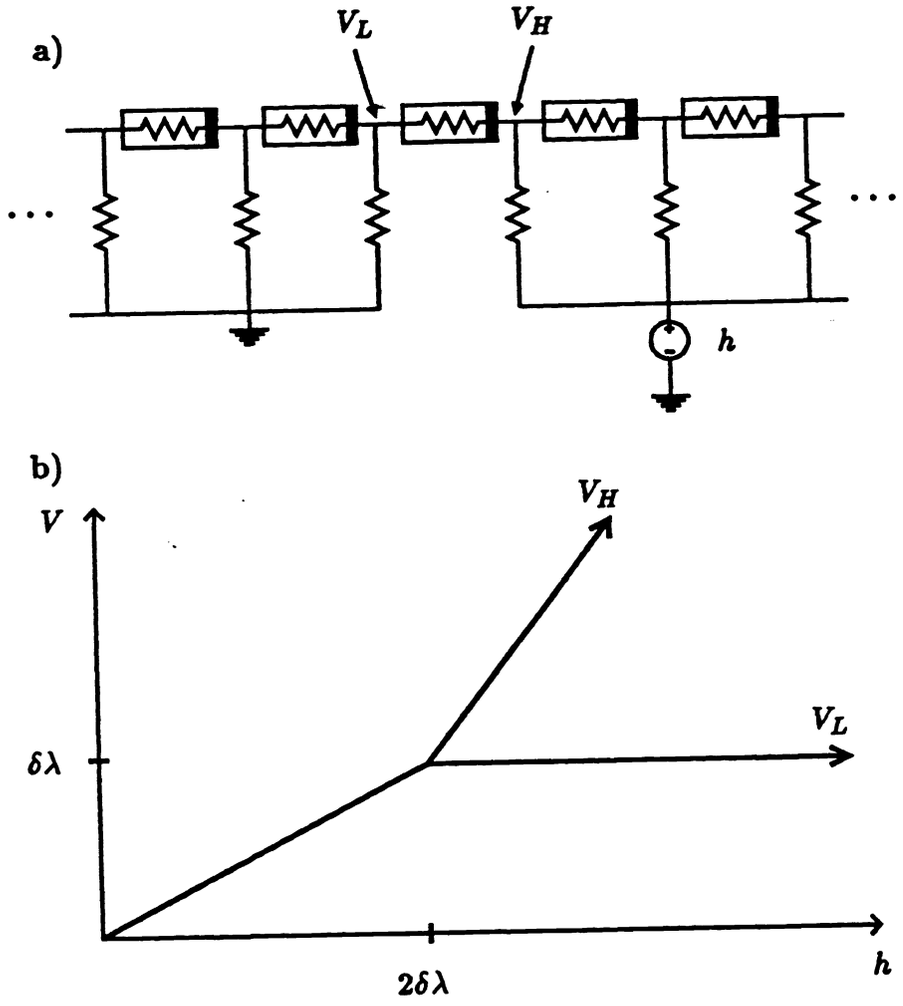

Figure 7: a) Schematic of step input of height h for the nonlinear network shown in Fig. 1. **b)** Plot of voltage V_H and V_L versus h assuming $\lambda \gg 1$. Inputs less than $2\delta\lambda$ are smoothed out since $V_H - V_L \approx 0$. Inputs steps larger than $2\delta\lambda$ are recovered as step edges with $V_H - V_L \approx h - 2\delta\lambda$.

Notice that the network does not recover the exact heights of input edges. Rather it subtracts a constant ($2\delta\lambda$) from the height of each input. This effect is noticeable in Fig. 5b where the unit step input has decreased slightly in magnitude. In two dimensions, this effect becomes more interesting. For regions with small area to circumference ratios, the step height will decrease by a larger amount. Typically, the exact values of the heights are less important than the location of the discontinuities. Furthermore, it would not be difficult to construct a two-stage network to recover the exact values of the step heights if desired. In this scheme a tiny-tanh network would control the switches on a second fuse network.

In retrospect, we have changed the nonlinear line-process computation to a simpler problem that can be solved by minimizing a convex energy. Notice that the network does not have an explicit representation of discontinuities. It merely subtracts a fixed voltage ($2\delta\lambda$) from the height of input step edges. If a binary discontinuity map is desired, a threshold must be chosen for an additional post-processing step. Unlike typical line-process energy minimization algorithms, the whole segmentation computation is accomplished without defining a binary threshold for edges.

4 CONCLUSION

A rigid piecewise-constant assumption about the world is unnecessarily restrictive. The continuous-time network simulated here could be used to implement a thin plate energy and segment based upon higher order discontinuities (Grimson, 1981; Terzopoulos, 1983; Blake and Zisserman, 1987; Harris, 1989; Liu and Harris, 1989). A successful extension of these methods will allow depth and intensity to be segmented into piecewise linear regions.

We have discussed a hardware resistive fuse network that segments inputs in real-time. We proposed a much simpler tiny-tanh network that performs similar computations in continuous-time. The paper illustrates two of the fundamental advantages of using analog circuits to experiment with vision algorithms. This new computational medium gives us the ability to deal with time explicitly and rapidly prototype nonlinear systems. The resistive fuse network is a highly nonlinear system which has forced us to think explicitly about dealing with time in our solution strategies. Continuation methods are awkward to implement in continuous-time fashion and so alternative methods are desirable. Through experimentation with analog hardware, we have discovered an algorithm that is useful for any sort of implementation—analog or digital.

ACKNOWLEDGEMENTS

This work was done in close cooperation with Christof Koch. Thanks to Carver Mead for making this research possible. All chips were fabricated through MOSIS with DARPA's support. This research was partially supported by NSF grant IST-8700064, a grant from the Office of Naval Research, by DDF-II funds from the Jet Propulsion Laboratory at the California Institute of Technology.

5 REFERENCES

Blake, A. (1989). Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**:2–12.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction.* Cambridge, MA: MIT Press.

Chua, L. O., Desoer, C. A., and Kuh, E. S. (1987). *Linear and Nonlinear Circuits.* New York: McGraw-Hill, pp. 23–34.

Geiger, D. and Girosi, F. (1989). Parallel and deterministic algorithms from MRF's: surface reconstruction and integration, MIT AI Memo 1114, May 1989.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**:721–741.

Grimson, W. E. L. (1981). *From Images to Surfaces.* Cambridge, MA: MIT Press.

Harris, J. G. (1989). An analog VLSI chip for thin plate surface interpolation. In *Neural Information Processing Systems I*, ed. D. Touretzky. Palo Alto: Morgan Kaufmann.

Harris, J. G., Koch, C., Luo, J. and Wyatt, J. (1989). Resistive fuses: analog hardware for detecting discontinuities in early vision. In: *Analog VLSI Implementations of Neural Systems*, Mead, C. and Ismail, M. eds., Kluwer, Norwell, MA.

Harris, J. G., Koch, C., and Luo, J. (1990a). A two-dimensional analog VLSI circuit for detecting discontinuities in early vision. *Science* **248**:1209-11.

Harris, J. G., Koch, C., Staats, E., and Luo, J. (1990b). Analog hardware for detecting discontinuities in early vision. *Internat. Journal of Comp. Vision* **4**:211-223.

Hildreth, E.C. (1984), *The Measurement of Visual Motion.* MIT Press, Cambridge, MA.

Horn, B.K.P. (1974). Determining lightness from an image. *Compt. Graph. Imag. Processing* **3**:277-299 (1974).

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artif. Intell.* **17**:185–203.

Horn, B. K. P. (1989). Parallel networks for machine vision. *Artif. Intell. Lab. Memo No.* **1071**, MIT, Cambridge, MA.

Hutchinson, J., Koch, C., Luo, J., and Mead, C. (1988). Computing motion using analog and binary

resistive networks. *IEEE Computer* **21**:52–63.

Koch, C., Marroquin, J., and Yuille, A. (1986). Analog "neuronal" networks in early vision. *Proc. Natl. Acad. Sci. USA* **83**:4263–4267.

Koch, C. (1989). Seeing chips: analog VLSI circuits for computer vision. *Neural Computation* **1**:184–200.

Liu, S. C. and Harris, J. G. (1989). Generalized smoothing networks in solving early vision problems. *Proc. IEEE Computer Vision and Pattern Recognition Conference*, San Diego, CA, June 1989, pp. 184-191.

Lumsdaine, A., Wyatt, J. and Elfadel, I. (1990), Nonlinear Analog Networks for Image Smoothing and Segmentation, Proc. 1990 *IEEE Internat. Symp. on Circuita and Systems*, New Orleans, LA, May 1990, pp. 987-991.

Mead, C. A. (1989). *Analog VLSI and Neural Systems.* Reading: Addison-Wesley.

Perona, P. and Malik, J. (1988). A network for multiscale image segmentation. *Proc. 1988 IEEE Int. Symp. on Circuits and Systems*, Espoo, Finland, June, pp. 2565–2568.

Poggio, T. and Koch, C. (1985). Ill-posed problems in early vision: from computational theory to analogue networks. *Proc. R. Soc. Lond. B* **226**:303–323.

Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature* **317**:314–319.

Poggio, T., Voorhees, H., and Yuille, A. (1986). A regularized solution to edge detection. Artif. Intell. Lab Memo **No. 833** (MIT, Cambridge).

Terzopoulos, D. (1983). Multilevel computational processes for visual surface reconstruction. *Comp. Vision Graph. Image Proc.* **24**:52–96.

Terzopoulos, D. (1986). Regularization of inverse problems involving discontinuities. *IEEE Trans. Pattern Anal. Machine Intell.* **8**:413–424.